710 James Robertson Pkwy
Nashville, TN 37243
Phone: (615) 741-2731

@TNedu
Tennessee Education
www.TN.gov/Education

CTSO Course Alignments: Programming and Logic II

Below you will find standards for the Programming and Logic II course aligned with competitive events from appropriate career and technical student organizations (CTSOs). Knowing the aligned events for your organization will allow you to have additional tools for teaching course standards, as well as increase student engagement and preparation in your CTSO activities. The final column recommends potential tools from other CTSO organizations. Even if your students are not participating in these organizations, available rubrics, tools, and materials can also add to the instructional resources at your disposal for best teaching your content.

Important to note: While the aligned activities below can be important tools in teaching course standards, it is important to note that events may not cover a standard in its entirety and should not be the sole instructional strategy used to address a standard.

	STANDARD	ALIGNED SkillsUSA & TSA COMPETITIVE EVENTS/PROGRAMS	OTHER POTENTIAL CTSO TOOLS & RESOURCES
1	Evaluate at least two software development environments (SDEs) that are tailored to different programming languages on the basis of their suitability for a range of programming tasks, ease of use, and how ubiquitous they are within the IT community. Document in an oral presentation the similarities and differences between the two, and the features that lend themselves to the chosen programming languages. For example, students assigned to code a basic database interface can compare the benefits and features of a freeware SDE such as *JDeveloper* and a commercial SDE like *Microsoft Visual Studio*. (TN Reading 1, 2, 5, 7, 9; TN Writing 2, 4, 6, 8, 9)	• **TSA**: Prepared Presentation	• **DECA:** Professional Selling Event • **HOSA**: Researched Persuasive Speaking, Prepared Speaking
2	Investigate the typical process around creating new software within a software development environment. Describe and furnish examples of the steps taken within the SDE to guarantee reliable output, from prototyping and authoring to deployment and debugging. (TN Reading 2, 3; TN Writing 2, 7, 8)	• **TSA**: Prepared Presentation	• **HOSA**: Researched Persuasive Speaking, Prepared Speaking

3	Administer the process of creating new software within a software development environment to manage the prototyping, authoring, revising, compiling, testing, deploying, and debugging of student-developed software. For example, for an object-oriented payroll program assignment (retrieving file data to produce a run of paychecks and paystubs for a small business), perform and document the steps taken within the SDE to ensure the reliable and accurate output of paychecks. (TN Reading 3, 4, 5; TN Writing 6, 7)		
4	Synthesize information from a range of sources (including original tests and simulations) to critique the features of different software development life cycles (agile, iterative, and sequential types). Using domain-specific terminology, explain to a technical audience the distinguishing features of each that make one more appropriate for certain types of applications. (TN Reading 2, 4, 9; TN Writing 2, 4, 9)	• **TSA**: Essays on Technology, Prepared Presentation	• **HOSA**: Researched Persuasive Speaking, Prepared Speaking
5	For a selected assignment or project involving the development of original software, choose and defend a strategy to follow for the program's development life cycle. At the completion of the assignment, offer recommendations for other environments and alternative strategies that could improve the development process. (TN Reading 3, 6, 8, 9; TN Writing 1, 4, 7, 9)		
6	Research common and best-practice techniques in programming analysis, design, and implementation. Drawing on model practices used by businesses and industry, employ analysis, design, and implementation techniques to satisfy a programming need, using an appropriate software lifecycle model. (TN Reading 2, 3, 5, 6; TN Writing 6, 8)		
7	Employ a requirement management tool during a program's development life cycle, documenting the evolving versions, storage attributes, system elements, status tracking, and access permissions afforded by the tool, as well as the successful attainment of the project vision. (TN Reading 3, 4, 5, 7, 9; TN Writing 4, 6, 9)		
8	For a given programming assignment, choose and defend a programming language with regard to the language's capabilities and suitability to task, availability, portability, maintainability, and cost. (TN Reading 3, 4, 5; TN Writing 1, 4)		
9	For the assignment outlined in standard 8, identify the method of data processing most appropriate for the task (e.g., batch, interactive, or event-driven). For example, a weekly payroll application would handle its data differently (i.e., batch processing) than a web-based search engine (i.e., interactive processing), and still differently than a microprocessor control program for a washing machine (i.e., event driven). (TN Reading 3, 4, 5, 6, 8)		

10	Define the specifications of the data management plan, including variables (naming, scope, and types), validation measures (to protect the data from corruption), and data handling (storing, input/output, and back-up). For example, programs handling historical temperature data would be best suited to floating point values stored in multidimensional arrays, written to permanent storage, and displayed with limited precision. (TN Reading 3, 4, 5)		
11	For a selected programming assignment involving an object-oriented language, design and define the classes, objects, properties, methods, and inheritance structures prior to the start of the development cycle. Revise the plan (modifications, additions, and subtractions) as needed throughout the development cycle. (TN Reading 2, 3, 4, 5, 7; TN Writing 5)		
12	For selected programming assignments, create, edit, and improve documentation for technical support intended for fellow programmers, including within the program code itself as well as within supplemental documents. For example, for a lawn sprinkler system microcontroller, the technical documentation would define the variables, functions and subroutines, and the critical events. (TN Reading 1, 3, 4, 5, 7; TN CSS Writing 2, 4, 5, 6, 10)		
13	For selected programming assignments, create, edit, and improve end-user documentation. End-user documentation would include how to interact with the user interface, the capabilities and limitations of the system, and the required conditions for successful operation. (TN Reading 1, 3, 4, 5, 7; TN CSS Writing 2, 4, 5, 6, 10)		• **FBLA:** Help Desk
14	Incorporate structured, object-oriented, and event-driven programming techniques that employ sequence, selection, and/or repetition (loops) to solve programming projects. (TN Reading 3, 5, 7; TN Writing 6, 7)		
15	For each programming task, consider and defend the choice of various programming approaches (such as data-driven or event-driven, top-down or bottom-up), citing examples from the syntax illustrating the chosen approach. (TN Reading 3; TN Writing 1, 4, 7, 8)		
16	Design and develop an app for a mobile computing device, using an online programming interface, such as AppMakr, BuzzTouch, Appsbar, PhoneGap, or AppYet. (TN Reading 2, 3, 4, 7; TN Writing 6, 7)		• **FBLA:** Mobile Application Development • **FFA:** Agricultural Communications
17	During the development, testing, and deployment of a new program, implement checks for data and procedure accuracy, correctness, currency, and relevance, making and documenting revisions where justified. (TN Reading 3; TN Writing 2, 4, 5, 6, 7, 10)		

18	Analyze the code written by another programmer to create a flowchart, suggesting points of confusion or generality in the program that could become problematic in future revisions. Cite specific examples in the code to support recommendations. (TN Reading 1, 2, 3, 4, 5, 6, 7, 8; TN Writing 1, 4, 6)		
19	Conduct quality testing of program code, striving for satisfactory results at four levels or perspectives: a) Unit (component/module level verifications) b) Integration (verifying the interfaces between components, adding one at a time) c) System (verifying that the whole package meets the requirements and specifications without corrupting other systems) d) Acceptance (customer satisfaction) (TN Reading 2, 3, 4, 5, 6)		
20	Design, manage, and develop a course-long programming project pre-approved by the instructor. The project will embody a variety of strategies and resources taught in this course, and require periodic reviews, status reports, and final project presentation. Use a software development environment to manage, document, test, deploy, and maintain the resources and assets of the finished project. (TN Reading 1, 2, 3, 4, 5, 6, 7, 8; TN Writing 1, 2, 4, 5, 6, 7, 8, 9)		
ALL	**CAN BE USED WITH ALL/MOST STANDARDS**	• **SkillsUSA**: Career Pathways Showcase, Job Skills Demonstration A, Job Skills Demonstration O, Prepared Speech, Extemporaneous Speaking, Chapter Display, Computer Programming • **TSA**: Software Development, Video Game Design	• **FBLA:** Mobile Application Development, Computer Game Simulation & Programming, Database Design & Application, Desktop Application Programming • **FCCLA**: Illustrated Talk, Career Investigation, Chapter in Review Display, Chapter in Review Portfolio